AI406L – Professional in AI Training Lab

Project 7: Chat Assistant + Pinecone

Database Website: https://www.giki.edu.pk

Registration No: 2022101

Submitted by: Anum Imran

Submitted to: Mr. Touqeer Abbas

Institute: Ghulam Ishaq Khan Institute of Engineering Sciences & Technology

Abstract

This project implements an **AI-powered Chat Assistant** designed to answer queries related to the **GIKI website**, including information about departments, faculty, research domains, and

academic programs. The workflow involves **web scraping**, **data preprocessing**, **vector embedding using Pinecone DB**, and **retrieval-based response generation** through a language model. The chatbot integrates **LangChain** and an **LLM (via Groq API)** to generate contextually accurate responses. Finally, the system is deployed on **AWS EC2** with **Nginx** for stability and scalability. This project demonstrates the practical use of **modern AI pipelines**, **vector databases**, and **cloud-based deployment** for real world educational applications.

1. Objectives

1. To scrape and preprocess GIKI's academic and research-related information.
2. To generate text embeddings and store them in Pinecone DB.
3. To implement a chatbot capable of providing relevant and context-aware responses to user queries.
4. To deploy the chatbot API on AWS EC2 using Nginx for cloud accessibility.
5. To ensure modular, reusable, and well-documented code for scalability and maintainability.

2. Tools and Technologies

Category	**Tools / Libraries**
Programming Language	Python 3.10
Libraries	requests, beautifulsoup4, selenium, dotenv, tqdm, json, langchain, pinecone, groq, pandas
Database	Pinecone Vector Database
Cloud Deployment	AWS EC2 / Nginx
Framework	LangChain + FastAPI
Environment	Virtual Environment (venv), VS Code

3. Methodology

The Chat Assistant + Pinecone DB project was developed following a structured pipeline from data scraping to deployment, ensuring efficiency and modularity throughout the process.

Step 1: Web Scraping

Data was extracted from the **official GIKI website** using both **BeautifulSoup (bs4)** and **Selenium**. Selenium handled dynamic pages while BeautifulSoup was used for static content. The scraper navigated through departmental, research, and academic sections, capturing relevant textual data which was then cleaned and stored in a structured JSON format.

Step 2: Index Creation and Data Upsertion

The preprocessed text data was segmented into smaller chunks and transformed into vector embeddings using a sentence embedding model. A **Pinecone vector index** was then created to store these embeddings efficiently. The embeddings were **upserted** (uploaded) into the Pinecone database. Retrieval tests confirmed that relevant chunks were returned accurately for test queries.

Step 3: Retrieval-Augmented Generation (RAG) Integration

After successful retrieval validation, a **RAG (Retrieval-Augmented Generation)** pipeline was constructed using **LangChain**. Retrieved text chunks were passed to a **Large Language Model (LLM)** via **Groq API**, along with a structured system prompt to ensure responses remained restricted to the retrieved GIKI content. This step ensured that the chatbot produced factual, contextually relevant answers.

Step 4: FastAPI Backend Setup

A **FastAPI** backend was created to handle API requests. Two main endpoints were developed: /query (for chat requests) and /health (for service monitoring). The API directly communicated with the LangChain RAG pipeline to deliver real-time query responses.

Step 5: Deployment

After local validation, the complete application was deployed on an **AWS EC2 instance**. The **FastAPI app** was hosted using **Uvicorn** and configured behind **Nginx** as a reverse proxy for stable, secure, and scalable performance. This allowed public access to the chatbot for evaluation and testing.

EC2 Deployment (Summary Steps)

- **Update system & install tools:** sudo apt update && sudo apt upgrade -y
  sudo apt install python3-pip python3-venv nginx git -y

- **Clone repo & enter project folder:** git clone https://github.com/anum-imm/giki-ai-lab project && cd project
- **Create and activate virtual environment:** python3 -m venv venv && source venv/bin/activate
- **Install dependencies:** pip install -r requirements.txt
- **Test FastAPI app locally:** uvicorn app:app --host 0.0.0.0 --port 8000
- **Configure Nginx reverse proxy**

  Restart: sudo systemctl restart nginx
- **Allow HTTP traffic:** sudo ufw allow 80
- **Run app in background:** uvicorn app:app --host 0.0.0.0 --port 8000 &
- **Access via browser:** http://< http://13.60.40.65>

(Nginx acted as a **bridge between the internet and the FastAPI server**, managing HTTP requests and ensuring the chatbot ran securely and reliably in a production environment.)

4. Results

- Successfully retrieved department and program-related information through chatbot queries.
- Achieved **semantic search-based query matching** using Pinecone vector embeddings.
- Deployed and hosted the chatbot on **AWS EC2** for live testing.
- Demonstrated accurate and context-aware query-to-response functionality for educational data.

5. Conclusion

This project demonstrates the seamless integration of **web scraping**, **vector database storage**, **retrieval-augmented generation**, and **cloud deployment** to build an intelligent chatbot. The system's ability to fetch relevant information from the GIKI website and respond intelligently highlights the practical potential of AI-driven automation in educational platforms. Future enhancements may include adding a frontend interface and expanding to multilingual support.

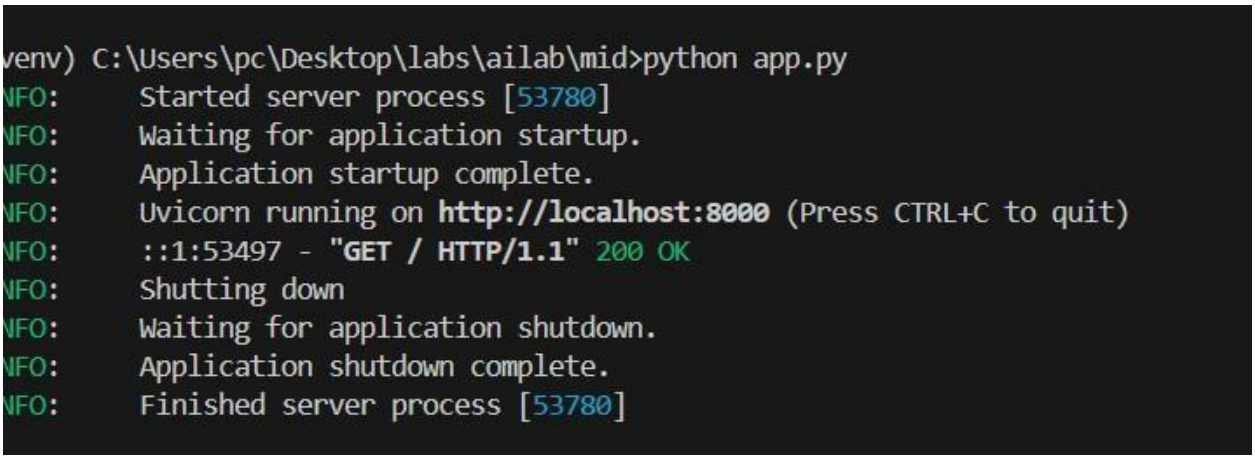
6. References

- Pinecone Documentation • LangChain Framework
- AWS EC2 Documentation
- Nginx Official Docs
- BeautifulSoup Documentation

Command to run it:

ssh -i "C:\Users\pc\Downloads\giki.pem" ubuntu@13.60.40.65

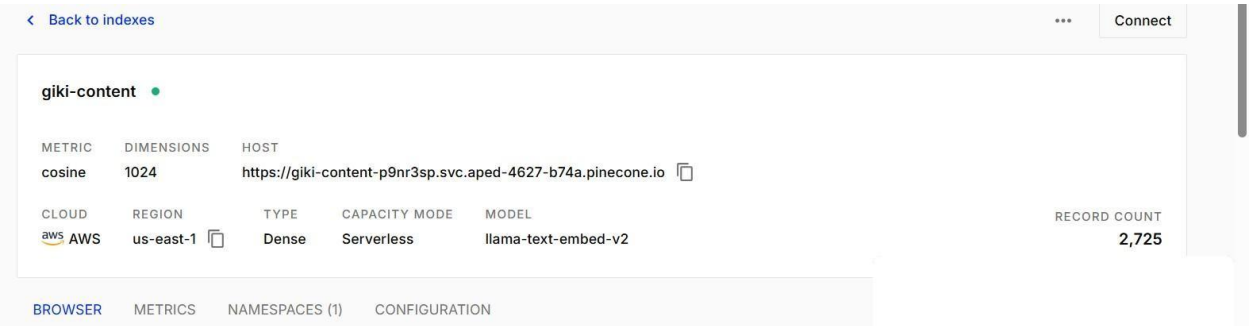
RESULTS:

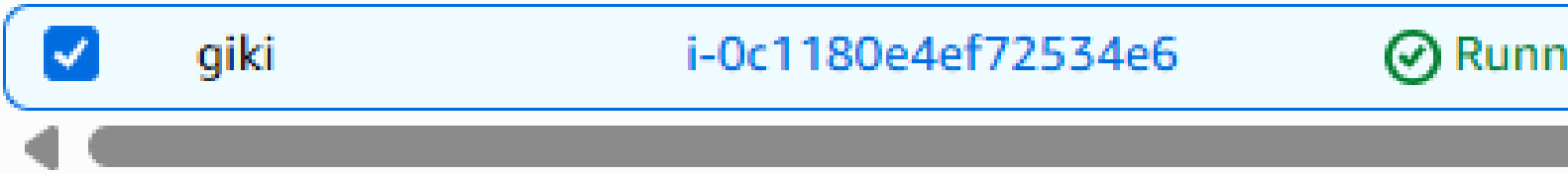


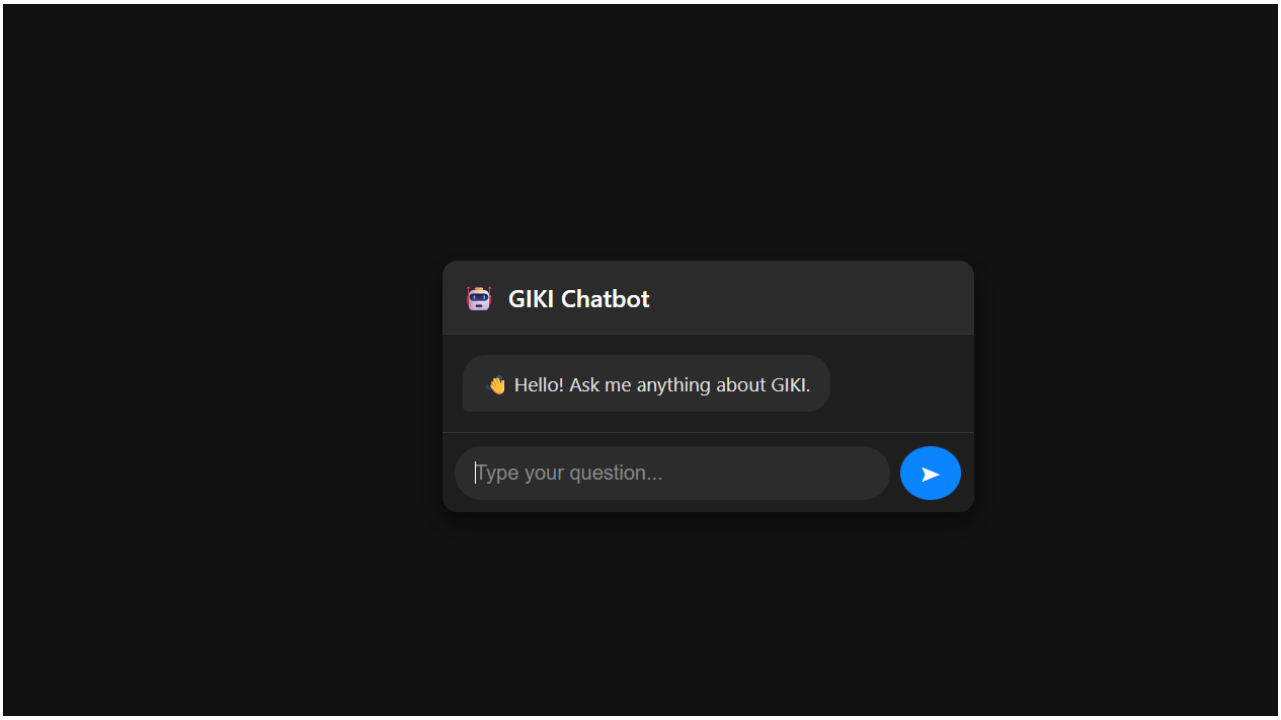
```
(venv) C:\Users\pc\Desktop\labs\ailab\mid>git init
Initialized empty Git repository in C:/Users/pc/Desktop/labs/ailab/mid/.git/

(venv) C:\Users\pc\Desktop\labs\ailab\mid>git add .

(venv) C:\Users\pc\Desktop\labs\ailab\mid>git status
On branch main
```






GIKI Chatbot
Hello! Ask me anything about GIKI.
Type your question...